Vulnerability Assessment Report of

Thales Drone Web Application

Jan 2022

**About L&T Technology Services:**

L&T Technology Services Limited (LTTS) is a global leader in Engineering and R&D (ER&D) services. With 399 patents filed for 51 of the Global Top 100 ER&D spenders. Our innovations speak for itself – World’s 1st Autonomous Welding Robot, Solar ‘Connectivity’ Drone, and the Smartest Campus in the World, to name a few. LTTS expertise in engineering design, product development, smart manufacturing, and digitalization touches every area of our lives. With 49 Innovation and R&D design centers globally, we specialize in disruptive technology spaces such as 5G, Artificial Intelligence, Collaborative Robots, Digital Factory, and Autonomous Transport. **LTTS is a publicly listed subsidiary of Larsen & Toubro Limited, the $18 billion Indian conglomerate operating in over 30 countries.**

Document History:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Var. Rel. No.** | **Release Date** | **Prepared By.**  **Prepared. Dt.** | **Reviewed By**  **Reviewed Dt.** | **Approved By**  **Approval Dt.** | **Remark/Revision Details** |
| 1.0 | Oct 2021 | Pallavi  Nikhil | Vishal |  |  |
| 24th Jan 2022 |  |  |

Table of Contents

[1. Overview of the project 5](#_Toc93939142)

[2. Vulnerabilities explained in detail 7](#_Toc93939143)

[**2.1 Security misconfiguration** 7](#_Toc93939144)

[**2.2 Privilege escalation supplier can download admin files** 8](#_Toc93939145)

[**2.3 Adding PO for different supplier** 11](#_Toc93939146)

[**2.4 Broken Authentication** 12](#_Toc93939147)

[**2.5 URL Traversal** 14](#_Toc93939148)

[**2.6 Supplier have access to delete admin files** 15](#_Toc93939149)

[**2.7 Supplier have access to download avenue files** 17](#_Toc93939150)

[**2.8 Test for user-accessible authentication history** 19](#_Toc93939151)

[**2.9 Insecure TLS Version Supported** 20](#_Toc93939152)

[**2.10 Insecure Direct Object References (IDOR)** 21](#_Toc93939153)

[**2.11 Information Disclosed through URL** 23](#_Toc93939154)

[**2.12 Privilege Escalation** 24](#_Toc93939155)

[**2.13 CORS Misconfiguration** 26](#_Toc93939156)

[**2.14 JSESSIONID manipulation** 27](#_Toc93939157)

[**2.15 Information Exposure of Ports and Services** 28](#_Toc93939158)

[**2.16 Weak Hashing Algorithms** 30](#_Toc93939159)

[**2.17 Insufficient Session Expiration** 31](#_Toc93939160)

[**2.18 No Content Security Policy** 32](#_Toc93939161)

[**2.19 Application Allows Concurrent Sessions** 33](#_Toc93939162)

[**2.20 No Multifactor Authentication (MFA)** 34](#_Toc93939163)

[**2.21 Server Banner Disclosure** 36](#_Toc93939164)

[**2.22 Logout CSRF** 37](#_Toc93939165)

[3. Abbreviation 39](#_Toc93939166)

[4. Appendix 39](#_Toc93939167)

# Overview of the project

L&T Technology Services (LTTS) security team has conducted Security Assessment for Thales Drone web application. The purpose of the assessment is to evaluate the security posture of the web application against common vulnerabilities.

**Objective of the security assessment:**

As a part of this engagement a holistic approach was taken to conduct the Vulnerability Assessment and Penetration Testing on Thales Drone web application. During the engagement High, Medium, and Low severity issues were identified with respect to Thales Drone web application.

**Approach**

The following approach was taken to make sure the target was assessed against known vulnerabilities from all possible security perspectives:

* Manual Vulnerability Assessment and Penetration Testing using OWASP TOP 10 for web application.

Some of the tools which were used are listed below:

|  |  |
| --- | --- |
| Target Application | Thales Drone |
| Browser | Chrome, Firefox |
| Tools | BURP, Nmap |

* + 2. Key Security Policies

OWASP top 10 listed vulnerabilities ware used as a reference framework. The following key security aspects were checked:

1. Broken Access Control

2. [Cryptographic Failures](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/)

3. [Injection](https://owasp.org/Top10/A03_2021-Injection/)

4. [Insecure Design](https://owasp.org/Top10/A04_2021-Insecure_Design/)

5. [Security Misconfiguration](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/)

6. [Vulnerable and Outdated Components](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/)

7. [Identification and Authentication Failures](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/)

8. [Software and Data Integrity Failures](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/)

9. [Security Logging and Monitoring Failures](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/)

10. [Server-Side Request Forgery](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/)

**Summary of Findings**

The graph below shows a summary of the number of vulnerabilities found for each impact level for the Application Security Assessment. Vulnerabilities found are addressed according to priority, findings, analysis, and recommendations from the assessment.

Table

Description automatically generated

# Vulnerabilities explained in detail

|  |  |  |  |
| --- | --- | --- | --- |
| **2.1 Security misconfiguration** | | | |
| **Impact** | High | **Risk Rating** | Critical |
| **Ease of Exploit** | Difficult |
| **Likelihood** | High |
| **Category** | Excessive Permissions to the user/supplier | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Both admin and supplier accounts are configured as administration on the OneLogin page. | | | |
| **Impact** | | | |
| * The supplier can enumerate the administrator portal information and supplier has equal and same access that admin has. * Not revoking supplier access after they are no longer needed may led to privilege escalation and other issues within the environment. | | | |
| **Recommendation** | | | |
| Configuration such that supplier/users on the application cannot manipulate their or an administrator setting. | | | |
| **How to recreate the Security defect** | | | |
| * Login into the application using admin credentials * And login with supplier credentials * Notice that both the admin and supplier are configured as administration. | | | |
| **Evidence** | | | |
| Graphical user interface, text, application, website  Description automatically generated  Graphical user interface, text, application  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.2 Privilege escalation supplier can download admin files** | | | |
| **Impact** | High | **Risk Rating** | Critical |
| **Ease of Exploit** | Moderate |
| **Likelihood** | High |
| **Category** | Privilege escalation | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Supplier can download restricted files that only admin should be able to download. | | | |
| **Impact** | | | |
| * The supplier can enumerate the administrator portal information. * Not revoking supplier access after they are no longer needed may led to privilege escalation and other issues within the environment. | | | |
| **Recommendation** | | | |
| * Implementation of Access control on file downloads which means only admin should have access to download any file from the application. * Resilient access control approach towards administrators and users. * It is highly recommended that there should be restriction on supplier and implement strict no access policy for supplier if they are trying to download the file from the application. | | | |
| **How to recreate the Security defect** | | | |
| * Login as supplier. * Convert PO number using base64 conversion. * Download any file. | | | |
| **Evidence** | | | |
| Graphical user interface, text, application  Description automatically generated  Graphical user interface, text, application  Description automatically generated  Graphical user interface, text, application  Description automatically generated  **Graphical user interface, text, application  Description automatically generated**  **Graphical user interface, text, application, Word  Description automatically generated**  **Graphical user interface, application  Description automatically generated** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.3 Adding PO for different supplier** | | | |
| **Impact** | High | **Risk Rating** | High |
| **Ease of Exploit** | Moderate |
| **Likelihood** | Medium |
| **Category** | Privilege escalation | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Supplier has access to add PO for different supplier | | | |
| **Impact** | | | |
| * The supplier can enumerate the administrator portal information. * Not revoking supplier access after they are no longer needed may led to privilege escalation and other issues within the environment. | | | |
| **Recommendation** | | | |
| It is highly recommended that there should be restriction on supplier and implement strict no access policy for supplier if they are trying to change the information from the application. | | | |
| **How to recreate the Security defect** | | | |
| * Login as supplier. * Suppliers have access to change the supplier ID * And we can see a message is successfully saved. | | | |
| **Evidence** | | | |
| Graphical user interface, text, application, email  Description automatically generated  **Graphical user interface, application  Description automatically generated with medium confidence**    **Graphical user interface, text, application, email  Description automatically generated**  **Graphical user interface, text, application  Description automatically generated** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.4 Broken Authentication** | | | |
| **Impact** | High | **Risk Rating** | High |
| **Ease of Exploit** | Moderate |
| **Likelihood** | Medium |
| **Category** | Broken Authentication | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| The application does not provide logout functionality. Applications traditionally provide functionality on each page which, when executed, destroys the user’s server-side session, and redirects them to the application login page. This helps prevent unauthorized access to a user’s account by limiting the amount of time the user’s session is active and provides legitimate users with a way to end their session in the event of session compromise. | | | |
| **Impact** | | | |
| Attacker can use previous used or available session token to login into application. | | | |
| **Recommendation** | | | |
| The user’s HTTP session should be terminated on the server immediately after a logout action is performed. It is important to note that simply deleting the cookie from the browser will not terminate the server session. The session must be invalidated at the server, using the HTTP container’s intrinsic session abandonment mechanism. | | | |
| **How to recreate the Security defect** | | | |
| * Click on sign-out * With the help of forward and backward one page, able to re-direct to the application even after sign-out. | | | |
| **Evidence** | | | |
| Graphical user interface, application  Description automatically generated  Graphical user interface  Description automatically generated with low confidence | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.5 URL Traversal** | | | |
| **Impact** | High | **Risk Rating** | High |
| **Ease of Exploit** | Easy |
| **Likelihood** | Medium |
| **Category** | Information disclosed | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Path traversal or Directory traversal is a security vulnerability that occurs when software uses attacker-controlled input to construct a pathname to a directory or file located outside of the restricted directory. An attacker might be able to read arbitrary files on the target system. | | | |
| **Impact** | | | |
| An attacker can gain access to sensitive and system information on the system, delete or modify files. The maximum impact depends on the functionality of the application. | | | |
| **Recommendation** | | | |
| * The most effective way to prevent file path traversal vulnerabilities is to avoid passing user-supplied input to filesystem APIs altogether. Many application functions that do this can be rewritten to deliver the same behaviour in a safer way. * The application should validate the user input before processing it. Ideally, the validation should compare against a whitelist of permitted values. If that isn't possible for the required functionality, then the validation should verify that the input contains only permitted content, such as purely alphanumeric characters. * After validating the supplied input, the application should append the input to the base directory and use a platform filesystem API to canonicalize the path. It should verify that the canonicalized path starts with the expected base directory. | | | |
| **How to recreate the Security defect** | | | |
| * Browse to – <https://thalesdrone-test.ltts.com/> * Navigate to edit profile URL * Notice that it navigates to the new next page contain information. | | | |
| **Evidence** | | | |
| A screenshot of a computer  Description automatically generated  Graphical user interface, application  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.6 Supplier have access to delete admin files** | | | |
| **Impact** | High | **Risk Rating** | High |
| **Ease of Exploit** | Moderate |
| **Likelihood** | Medium |
| **Category** | Privilege escalation | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| During the assessment, it was observed that Supplier can delete files that only admin should be able to download or delete. | | | |
| **Impact** | | | |
| * The supplier can enumerate the administrator portal information. * Not revoking supplier access after they are no longer needed may led to privilege escalation and other issues within the environment. | | | |
| **Recommendation** | | | |
| * Implementation of Access control on file downloads which means only admin should have access to download any file from the application. * Resilient access control approach towards administrators and users. * It is highly recommended that there should be restriction on supplier and implement strict no access policy for supplier if they are trying to download the file from the application. | | | |
| **How to recreate the Security defect** | | | |
| * Login as supplier. * change the URL by navigating to CRM, notice that supplier have access to delete the file and able to delete the file successfully. | | | |
| **Evidence** | | | |
| A screenshot of a computer  Description automatically generated  Graphical user interface, text, application  Description automatically generated  Graphical user interface, application  Description automatically generated  Graphical user interface, text, application  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.7 Supplier have access to download avenue files** | | | |
| **Impact** | High | **Risk Rating** | High |
| **Ease of Exploit** | Moderate |
| **Likelihood** | High |
| **Category** | Privilege Escalation | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| During the assessment, it was observed that Supplier can download avenue files that only admin should be able to download. | | | |
| **Impact** | | | |
| * The supplier can enumerate the administrator portal information. * Not revoking supplier access after they are no longer needed may led to privilege escalation and other issues within the environment. | | | |
| **Recommendation** | | | |
| * Implementation of Access control on file downloads which means only admin should have access to download any file from the application. * Resilient access control approach towards administrators and users. * It is highly recommended that there should be restriction on supplier and implement strict no access policy for supplier if they are trying to download the file from the application. | | | |
| **How to recreate the Security defect** | | | |
| * Login as supplier * Change the URL by navigating to avenue, notice that supplier have access to download which is only been accessible to admin. | | | |
| **Evidence** | | | |
| Graphical user interface, application  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.8 Test for user-accessible authentication history** | | | |
| **Impact** | High | **Risk Rating** | High |
| **Ease of Exploit** | Moderate |
| **Likelihood** | Medium |
| **Category** | Privilege escalation | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| During the assessment, it was observed that even after changing the password, user can land up in the application through browser history without giving the credentials. | | | |
| **Impact** | | | |
| Attacker can used History URL to gain access to application as it doesn't need any credentials. | | | |
| **Recommendation** | | | |
| The user’s HTTP session should be terminated on the server immediately after password changed action has been performed. | | | |
| **How to recreate the Security defect** | | | |
| * From the login page change the password of Supplier/admin * Go to History and click on the previously used session * Notice that you’re able to log-in into the application without entering the credentials. | | | |
| **Evidence** | | | |
| Graphical user interface, text, application  Description automatically generated  Graphical user interface, application  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.9 Insecure TLS Version Supported** | | | |
| **Impact** | High | **Risk Rating** | High |
| **Ease of Exploit** | Easy |
| **Likelihood** | Medium |
| **Category** | Inadequate Encryption Strength. | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| "The server-side SSL/TLS endpoint is configured to allow weak SSL/TLS cipher suites. These cipher suites have proven cryptographic flaws that can allow an attacker to decrypt or modify traffic. These weak cipher suites include the following:  1. Cipher suites that use RC4 for encryption; RC4 contains several known weaknesses. An attacker that can gather many ciphertexts that contain the same plaintext encrypted using different keys may be able to recover the plaintext (the typical target of this attack would be a cookie that is sent in multiple SSL/TLS sessions).  2. Cipher suites that use block ciphers (e.g., AES, 3DES) in CBC mode; these are vulnerable to the BEAST attack if SSL 3.0 or TLS 1.0 are supported. Even if newer versions of TLS are also supported by the server, older client software might establish SSL 3.0 or TLS 1.0 connections. Additionally, an attacker may be able to use the POODLE attack to downgrade the connection to SSL 3.0 or TLS 1.0 even if both the client and the server support newer versions of TLS. BEAST allows an attacker on the same network as an end user, who can inject code into any site open in the user's browser, to decrypt cookies (or other sensitive data that is part of each request) for the vulnerable site.  3. Cipher suites that use the DES block cipher; these cipher suites use 56-bit keys for encryption, which are brute-forceable.  4. EXPORT-strength cipher suites use encryption key lengths that are vulnerable to brute force attacks (40 bits or 56 bits for symmetric key algorithms and 512 bits for the RSA asymmetric key algorithm). Note that even if stronger algorithms are preferred by the server, having EXPORT-strength cipher suites enabled is dangerous due to attacks like FREAK and Logjam.  5. Cipher suites with NULL for encryption do not offer any encryption for the connection.  6. Anonymous Diffie-Hellman cipher suites do not authenticate the endpoints and are vulnerable to man-in-the-middle attackers that can perform active attacks to read/modify traffic." | | | |
| **Impact** | | | |
| A server-side SSL/TLS endpoint that supports weak ciphers could allow an attacker to read or modify traffic sent in SSL/TLS connections with that endpoint. | | | |
| **Recommendation** | | | |
| The server-side TLS endpoint's configuration should be updated to allow only TLSv1.2 connections with cipher suites that use:  1. Ephemeral Diffie-Hellman for key exchange (optionally, allow RSA for key exchange if necessary for supporting some clients)  2. Block ciphers with key lengths of at least 128 bits (AES-128 and AES-256; optionally allow 3DES with 112-bit keys if necessary for supporting some clients)  3. Block ciphers in GCM mode. Note: If CBC mode must be allowed for supporting some clients, use only CBC mode cipher suites that use the SHA2 family of hash functions (SHA256, SHA384, SHA512)  Note that all modern browsers support TLSv1.2. | | | |
| **How to recreate the Security defect** | | | |
| * Use Kali Linux/Ubuntu and run the Nmap NSE scripts. * Use the script for SSL\_ENUM\_CIPHERS. | | | |
| **Evidence** | | | |
| **Text  Description automatically generated** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.10 Insecure Direct Object References (IDOR)** | | | |
| **Impact** | High | **Risk Rating** | High |
| **Ease of Exploit** | Easy |
| **Likelihood** | Medium |
| **Category** | Improper Privilege Management | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Insecure direct object references (IDOR) are a cybersecurity issue that occurs when a web application developer uses an identifier for direct access to an internal implementation object but provides no additional access control and/or authorization checks. For example, an IDOR vulnerability would happen if the curl of a transaction could be changed through client-side user input to show unauthorized data of another transaction. | | | |
| **Impact** | | | |
| Exposure of Confidential Information: When the attacker will have control over your account via this vulnerability, it is obvious that an attacker will be able to come across your personal information. | | | |
| **Recommendation** | | | |
| • Developers should avoid displaying private object references such as keys or file names.  • Validation of Parameters should be properly implemented.  • Verification of all the Referenced objects should be done.  • Tokens should be generated in such a way that it should only be mapped to the user and should not be public. | | | |
| **How to recreate the Security defect** | | | |
| * Login to the web application. * Capture the user profile request with burp suite. * Forward the request to intruder. * Give payloads (1000-4000 numbers) in the Id position * After successful attack, we can get other user profile details. | | | |
| **Evidence** | | | |
| **Graphical user interface, text, application, Word  Description automatically generated** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.11 Information Disclosed through URL** | | | |
| **Impact** | Medium | **Risk Rating** | Medium |
| **Ease of Exploit** | Difficult |
| **Likelihood** | Medium |
| **Category** | Improper Privilege Management | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Login process communicates excessive information using JWT and the URL contain sensitive data so, if the URL has been shared with the unauthorized person, then it may lead to information disclosed because if the URL has been copied to the browser, then it’s not asking for the credentials it’s directly redirecting to the one’s account. | | | |
| **Impact** | | | |
| Exposure of Confidential Information: When the attacker will have control over your account via this vulnerability, it is obvious that an attacker will be able to come across the personal information and will have full access to view, modify, delete the data present in the application. | | | |
| **Recommendation** | | | |
| * Developers should avoid displaying private object references such as keys or file names. * Tokens should be generated in such a way that it should only be mapped to the user and should not be public. * Validation of Parameters should be properly implemented. * Verification of all the Referenced objects should be done. | | | |
| **How to recreate the Security defect** | | | |
| * Login into the application * Copy the URL after login * Then paste the URL in same or different browser * Notice that you can login to the application with any credentials. | | | |
| **Evidence** | | | |
| **Text  Description automatically generated**  **A screenshot of a computer  Description automatically generated** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.12 Privilege Escalation** | | | |
| **Impact** | Medium | **Risk Rating** | Medium |
| **Ease of Exploit** | Moderate |
| **Likelihood** | Medium |
| **Category** | Privilege escalation | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| During the assessment, it was observed that supplier and admin can use valid authentication token and gain unauthorized access to the application. | | | |
| **Impact** | | | |
| * The supplier can enumerate the administrator portal information. * Not revoking supplier access after they are no longer needed may led to privilege escalation and other issues within the environment. | | | |
| **Recommendation** | | | |
| * Resilient access control approach towards administrators and users. * It is highly recommended that there should be restriction on supplier and implement strict no access policy for users trying to change the data in the application. | | | |
| **How to recreate the Security defect** | | | |
| * Login as supplier. * Send the request in repeater then, Logout from supplier. * Then Login with admin credentials and change the authorized token from admin to supplier’s token and edit the config value. Notice that config value is been changed successfully. | | | |
| **Evidence** | | | |
| Graphical user interface, application, Word  Description automatically generated  A screenshot of a computer  Description automatically generated  Graphical user interface, application  Description automatically generated  Graphical user interface, text, application, Word  Description automatically generated  Graphical user interface, text, application  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.13 CORS Misconfiguration** | | | |
| **Impact** | Medium | **Risk Rating** | Medium |
| **Ease of Exploit** | Moderate |
| **Likelihood** | Medium |
| **Category** | Permissive Cross-domain Policy with Untrusted Domains | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Access-allow control attribute is incorrectly set using wildcards such as (\*) under which domains can request resources.  It enables controlled access to resources which are outside of the given domain. It adds flexibility to Same Origin Policy (SOP) | | | |
| **Impact** | | | |
| This is usually set as default, which means any domain can access resources on this site.  Able to steal confidential and sensitive information. | | | |
| **Recommendation** | | | |
| * Set the Access-allow control header to validated and whitelisted websites * To implement CORS securely, you need to associate a validation list with Access-Control Allow-Origin that identifies which specific domains can access resources. Then your application can validate against this list when a domain requests access. | | | |
| **How to recreate the Security defect** | | | |
| * Browse to - <https://thalesdrone-test.ltts.com/> * Capture traffic observe response using burp suite * Change origin in request to any URL and observe response | | | |
| **Evidence** | | | |
| **Graphical user interface, text, application  Description automatically generated** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.14 JSESSIONID manipulation** | | | |
| **Impact** | Medium | **Risk Rating** | Medium |
| **Ease of Exploit** | Difficult |
| **Likelihood** | Medium |
| **Category** | Authentication bypass | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Random value set for JSESSIONID no server checks in place to authenticate the ID | | | |
| **Impact** | | | |
| Authentication bypass vulnerability could allow attackers to perform various malicious operations by bypassing the device authentication mechanism. If a hacker can get access to a valid session ID, then they can carry out authentication bypassing by doing a session hijack attack – essentially providing the server with the session ID of someone who has already been authenticated, thereby impersonating them. | | | |
| **Recommendation** | | | |
| * They suggest ensuring that user session IDs and cookies are encrypted. * It is best to have a secure and strong authentication policy in place. * It is best to ensure all systems, folders, apps, are password protected. * It is suggested to not expose authentication protocol in the client-side web browser script. * It is recommended to validate all user input on the server side. * It further recommended sending all cookies and session data over an encrypted channel. | | | |
| **How to recreate the Security defect** | | | |
| * Login to the application * Capture the request in burp * Manipulate the Session ID * Observe the response | | | |
| **Evidence** | | | |
| Graphical user interface, text, application  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.15 Information Exposure of Ports and Services** | | | |
| **Impact** | Medium | **Risk Rating** | Medium |
| **Ease of Exploit** | Moderate |
| **Likelihood** | Medium |
| **Category** | Security Misconfiguration | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Security misconfiguration can happen at any level of an application stack, including the network services, web server, application server, database, frameworks, or storage. It is observed that the application displays open ports and servers while performing manual penetration testing. Such flaws frequently give attackers unauthorized access to some system data or functionality. | | | |
| **Impact** | | | |
| If attacker intercept network and get user's credentials that may lead user's account get compromised. | | | |
| **Recommendation** | | | |
| Configure all external and internal network devices and services supporting sensitive information to utilize an encrypted protocol for authentication and transmission of data. Configure services and devices to encrypt their network traffic with SSL/TLS, SSH, or IPSEC. If the service natively supports some form of traffic encryption, enable that feature of the service. A common example of encrypting basic network service traffic is the use of HTTPS (SSL/TLS) to encrypt standard HTTP traffic. | | | |
| **How to recreate the Security defect** | | | |
| * Scan with NMAP with required target. * Found open ports. | | | |
| **Evidence** | | | |
| **Text  Description automatically generated**  **Text  Description automatically generated** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.16 Weak Hashing Algorithms** | | | |
| **Impact** | Medium | **Risk Rating** | Medium |
| **Ease of Exploit** | Easy |
| **Likelihood** | Medium |
| **Category** | Encryption and Authentication | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| Weak hashing algorithms such as MD2, MD5 and SHA-1 are known to be susceptible to collision attacks. Collisions in hashing algorithms occur when multiple data sets can be specifically constructed to produce the same resulting digest value. | | | |
| **Impact** | | | |
| Use of weak hashing algorithms may result in sensitive data exposure, key leakage, broken authentication, insecure session, and spoofing attack. | | | |
| **Recommendation** | | | |
| Employ one of the SHA-2 or SHA-3 cryptographic hash functions whenever it is important so that an attacker will not be able to generate multiple pieces of data that have the same hash. Even though no practical attack against SHA-1 exists yet, it is advisable to stop using it wherever possible to avoid future problems. | | | |
| **How to recreate the Security defect** | | | |
| * Use Kali Linux/Ubuntu and run the Nmap scan on port 443 | | | |
| **Evidence** | | | |
| Text  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.17** **Insufficient Session Expiration** | | | |
| **Impact** | Medium | **Risk Rating** | Medium |
| **Ease of Exploit** | Easy |
| **Likelihood** | Medium |
| **Category** | Insufficient session Timeout | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| An attacker's ability to hijack a victim's session increases proportionally with the amount of time an idle user's session remains valid. Once a valid session identifier is obtained, the attacker can impersonate the victim in the application, performing any functionality and accessing any data made available to the victim. Longer session timeouts also prevent server memory from being released, resulting in potential denial of service conditions when an attacker initiates many sessions in a sufficiently short period. | | | |
| **Impact** | | | |
| * Insufficient Session timeout increases a Web site exposure to attacks that steal or reuse user's session identifiers. * Cookie Hijacking is possible, application integrity can be compromised. * Once a valid session identifier is obtained, the attacker can impersonate the victim in the application. | | | |
| **Recommendation** | | | |
| Terminate the user's session server-side after a sufficiently short idle period. When the user makes further requests using the expired session, they should be redirected to a splash page or the login pages. In addition, client-side code should track session idle time and automatically redirect the user to a splash page or the login page after a certain period of client-side inactivity has passed. No prior authenticated user data or functionality should continue to be displayed after the timeout occurs.  Determine a session timeout duration that sufficiently protects end users and the application while maintaining system usability. Session timeouts of 30-120 minutes are common for most web applications and vary depending on the sensitivity of the information available during each session. Various standards organizations and government entities also typically recommend organization-defined timeouts or call for an idle timeout of 30-120 minutes:   * PCI DSS v3.2 section 8.1.8 states, "If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the terminal or session." * U.S. CNSS - CNSSI No. 1253 section AC-11 states, "Session Lock ... not to exceed 30 minutes" * NIST SP800-53 section AC-11 states, "...Prevents further access to the system by initiating a   session lock after [Assignment: organization-defined period of inactivity or upon receiving a request from a user". | | | |
| **How to recreate the Security defect** | | | |
| * Login to the application * Idle the application up to 30-60 minutes * Can access the application even after 30-60 minutes | | | |
| **Evidence** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.18 No Content Security Policy** | | | |
| **Impact** | Low | **Risk Rating** | Low |
| **Ease of Exploit** | Difficult |
| **Likelihood** | Medium |
| **Category** | Improper Restriction of Rendered UI Layers or Frames | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| During the HTTP traffic analysis of the Thales Drone web interface, it was observed that the server does not support Content Security Policy. Content Security Policy is a standard that helps protect against various content injection attacks like cross site scripting. While the victim is interacting with seemingly harmless web pages. | | | |
| **Impact** | | | |
| Without a Content Security Policy, an attacker can perform content injection attacks if data from the service is displayed in a browser | | | |
| **Recommendation** | | | |
| Enabling Content Security Policy response header to all HTTP server responses helps in preventing content injection attacks. While adding Content Security Policy it must be set correctly specifying the locations from which content can be loaded. Content-Security-Policy: <Policy-directive>; | | | |
| **How to recreate the Security defect** | | | |
| * + Browse to - <https://csp-evaluator.withgoogle.com/>   + Enter the URL - <https://thalesdrone-test.ltts.com/>   + Click on check CSP | | | |
| **Evidence** | | | |
| Graphical user interface, text, application  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.19 Application Allows Concurrent Sessions** | | | |
| **Impact** | Low | **Risk Rating** | Low |
| **Ease of Exploit** | Difficult |
| **Likelihood** | Low |
| **Category** | Manage User Sessions | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| The application allows concurrent sessions; multiple users can login to the application simultaneously with the same user credentials. | | | |
| **Impact** | | | |
| Attacker can make victim's account active with the same username and password. | | | |
| **Recommendation** | | | |
| The application should only allow a user to establish a single session with a particular set of credentials at a time. Once that session has been established, subsequent attempts to login using those credentials should either be denied or existing sessions should be terminated, depending on business needs.  If concurrent sessions are required for business purposes, additional session management features must be provided to ensure that all end users are made aware of multiple sessions. Such features include allowing end users to view all current sessions, prompting users when a new session is created, and providing users the ability to terminate unwanted sessions. Additionally, when allowing concurrent sessions, it is recommended that users are notified that their credentials were used to establish a new session, including the time and IP from which the session was established | | | |
| **How to recreate the Security defect** | | | |
| * Login into the application * Capture traffic observe response * Again, Login in the new tab with same credential as before * Notice that we multiple users can login to the application simultaneously with the same user credentials. | | | |
| **Evidence** | | | |
| **A screenshot of a computer  Description automatically generated** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.20 No Multifactor Authentication (MFA)** | | | |
| **Impact** | Medium | **Risk Rating** | Low |
| **Ease of Exploit** | Easy |
| **Likelihood** | Low |
| **Category** | Use of Single-factor Authentication | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| The application uses single-factor authentication to authenticate privileged users to the system. Single-factor authentication refers to the use of a single component to identify an end user to an application or system. The factor provided may be something the user knows, something the user is, or something the user has. Each of these options provides their own set of advantages and risks when used for authentication:  • "Something you know", such as a user-defined password, may be easily created and changed when necessary. Authentication factors derived from the end user must have some degree to be managed by the user themselves, leaving the known secret's security up to them. This can result in the secret being forgotten or exposed through a breach of a separate system that holds or uses the same known secret.  • "Something you are", such as a fingerprint, provides an end user with a constant factor that cannot be easily acquired or mimicked by an attacker. While this initially provides a strong barrier to entry and will always be with the end user, a single breach could leave the attribute used for authentication useless as it cannot be updated.  • "Something you have", such as a hardware token, can be managed from a central source and is configured to constantly update, removing responsibility for the known secret from the user. However, this transition of the knowledge base may hinder the application's accessibility if the device is not always at hand. | | | |
| **Impact** | | | |
| If an attacker compromises the authentication mechanism (e.g., a victim's account password), they will have full access to functionality and data normally only available to the victim. | | | |
| **Recommendation** | | | |
| Multi-factor authentication should be implemented and enforced for externally accessible applications containing sensitive data or functionality. Multi-factor authentication is built upon the combination of two or more components that can prove a user's identity to the application. This provides an additional layer of security as it is assumed that an unauthorized attacker will not be able to supply both factors required for authentication. The factors required by the application should be a combination of at least two distinct factors from the following:  • Something the user knows  • Something the user is  • Something the user has  For example, a common multi-factor authentication mechanism requires a user to provide a password they have created (something they know), as well as a value from a hardware token (something they have). If an attacker can compromise a user's password, they will still not have access to the hardware token and will not be able to gain access to the system.  Note: requiring two or more pieces of information for authentication that fall under the same factor category does not provide true multi-factor authentication. For example, a user's password and the answer to their security question are both something the user knows. Requiring both during authentication does not represent true multi-factor authentication. | | | |
| **How to recreate the Security defect** | | | |
| * + Browse application   + Capture traffic observe response   + Found the response without set cookie flag | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.21 Server Banner Disclosure** | | | |
| **Impact** | Low | **Risk Rating** | Info | |
| **Ease of Exploit** | Moderate |
| **Likelihood** | Low |
| **Category** | Information Disclosure | | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | | |
| **Description** | | | | |
| While performing vulnerability assessment and penetration testing, it was observed that verbose server information of sent in the responses from the server. The information is commonly included in the server response headers and can disclose information like server name, type, and version number. | | | | |
| **Impact** | | | | |
| Server banners provide additional information that allows an attacker to perform targeted attacks to the specific technology stack in use by the application and underlying infrastructure. | | | | |
| **Recommendation** | | | | |
| * Server information should be removed from all HTTP responses. This can be performed by modifying the server's configuration files or through the use and configuration of a web application firewall. * It is recommended to use generic error message response from server, so that server banner is disclosed in the error message response from the server. | | | | |
| **How to recreate the Security defect** | | | | |
| Login to the application | | | | |
| **Evidence** | | | | |
| Text  Description automatically generated | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.22 Logout CSRF** | | | |
| **Impact** | Low | **Risk Rating** | Info |
| **Ease of Exploit** | Moderate |
| **Likelihood** | Low |
| **Category** | Improper Authentication | | |
| **URL/Impacted system** | <https://thalesdrone-test.ltts.com/> | | |
| **Description** | | | |
| A Cross-Site Request Forgery (CSRF) vulnerability occurs when the application fails to ensure that requests received by the server originated from pages served by the application. A page from the malicious website sends an HTTP request to the application with data supplied by the attacker. The HTTP request causes the victim's browser to automatically send the user's session cookie along with the request. The application processes the request as though the request had been made from one of the application's pages. | | | |
| **Impact** | | | |
| The CSRF takes advantage of the fact that logout URLs are typically encrypted. The Logout CSRF does not offer a significant danger, although it is inconvenient for users. | | | |
| **Recommendation** | | | |
| Preventing a CSRF vulnerability requires adding a non-predictable parameter to requests that performs any transaction that modifies the application's data or internal state. This parameter must be part of the form and not stored in the cookie. The easiest way to add a non-predictable parameter is to use a secure hash function, such as SHA-2, to hash the user's session ID. This parameter is referred to as a "form-authenticator" because this parameter is checked on the server to ensure that this is a legitimate request from the application. The application includes the form-authenticator when a legitimate page is generated (e.g., as a hidden field). When the form is then submitted, the application verifies that the form-authenticator provided from the form matches the one issued when the page was generated. If they match, the request can be processed; otherwise, the application issues an error indicating the request is invalid. | | | |
| **How to recreate the Security defect** | | | |
| * Browse to Thales page * Capture the Logout request in burp. Create a CSRF Proof of concept on logout page. | | | |
| **Evidence** | | | |
| Graphical user interface, text, application, email  Description automatically generated  Graphical user interface, text, application  Description automatically generated  Graphical user interface, application, Word  Description automatically generated  Graphical user interface, application  Description automatically generated | | | |

# Abbreviation

|  |  |
| --- | --- |
| APP | Application |
| HTML | Hyper Text Mark-up Language |
| HTTP(S) | Hypertext transfer protocol (Secured) |
| Pg. | Page |
| TLS | Transport Layer Security |
| SSL | Secure Sockets Layer |
| IP | Internet Protocol |
| LTTS | Larsen & Toubro Technology Services |
| SOP | Same Origin Policy |
| OWASP | Open Web Application Security Project |
| VAPT | Vulnerability Assessment and Penetration testing |
| CSP | Content Security Policy |
| CORS | Cross Origin Resource Sharing |
| IDOR | Insecure direct object references |
| MFA | No Multifactor Authentication |
| URL | Uniform Resource locator |
| XSS | Cross-Site Scripting |
| XXE | XML External Entities |
| SQL | Structured Query Language |

# Appendix

Vulnerability scan reports.

|  |
| --- |
|  |

|  |
| --- |
|  |